

**DEVELOPMENT OF HOME AUTOMATION APPLICATION WITH FIREBASE AND NODE MCU**

**BY**

**IBRAHIM IBRAHIM SHEHU**

**U14CO1020**

**Department of Computer Engineering,**

**Ahmadu Bello University, Zaria**

**2019**



**DEVELOPMENT OF HOME AUTOMATION APPLICATION WITH FIREBASE AND NODE MCU**

**BY**

**IBRAHIM IBRAHIM SHEU**

**U14CO1020**

**Ibrahimshehu73700@gmail.com**

**Supervised by**

**Dr. T H SIKIRU**

**A PROJECT SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING, AHMADU BELLO UNIVERSITY, ZARIA – NIGERIA IN PARTIAL FULFILMENT OF THE REQUIRMENTS FOR THE AWARD OF THE DEGREE OF BACHELOR OF ENGINEERING(B.ENG) IN COMPUTER ENGINEERING**

**OCTOBER, 2019**

**DECLARATION**

I , Ibrahim Ibrahim Shehu, hereby declare that this project titled Home automation application with Firebase and Node MCU has been carried out by me under the supervision of Dr. H Sikiru. It has been presented for award of any degree in any institution. All sources of information are specifically acknowledged by means of reference.

..………………………… …….……………………

Signature Date

**CERTIFICATION**

This project entitled Home automation application with firebase and node mcu meets the requirements governing the award of the degree of Bachelor of Science in Computer Science and is approved for its contribution to knowledge and literary presentation.

……………………………………… …………………………….

Dr.T H Sikiru Date

Supervisor

…………………………………….. ….………………………….

Prof. M.B Muazu Date

Head of Department

….……………………………… …….………………………..

Project cordinator

Dr H Bello Salau Date

**ACKNOWLEDMENT**

I wish to use this opportunity to show my sincere appreciation and gratitude to the Almighty Allah (SWT) for giving me the grace to finish this study, and I will forever remain thankful to him for his guidance in every step I took and in every decision I had to make and for granting me the opportunity, good health, courage, reasoning.

**ABSTRACT**

Home Automation can be used to automate various activities in home and office by interfacing with traditional electrical and mechanical devices. Using Node MCU for home automation has numerous advantages such as low power consumption, low cost and powerful hardware with a very small footprint. In this project we are proposing a system to intuitively control and monitor electrical components at home or office using a real time control system using Node MCU and Android and firebase. The proposed system can effectively reduce waste of energy, save cost and add convenience to daily life. The system uses an Android application as the user interface and the Node MCU is used to control and monitor electrical components via GPIO pins and firebase to serve as database. The system is inexpensive, secure, easy to implement and uses open-source technologies.

Table of Contents

[CHAPTER ONE](#_Toc290_WPSOffice_Level1) [9](#_Toc290_WPSOffice_Level1)

[INTRODUCTION](#_Toc4314_WPSOffice_Level1) [9](#_Toc4314_WPSOffice_Level1)

[1.1 Background of study](#_Toc1907_WPSOffice_Level1) [9](#_Toc1907_WPSOffice_Level1)

[1.2 Motivation](#_Toc13757_WPSOffice_Level1) [9](#_Toc13757_WPSOffice_Level1)

[1.3 Problem statement](#_Toc12940_WPSOffice_Level1) [10](#_Toc12940_WPSOffice_Level1)

[1.4 Aims and Objectives](#_Toc11083_WPSOffice_Level1) [10](#_Toc11083_WPSOffice_Level1)

[1.5 Project organization](#_Toc8441_WPSOffice_Level1) [10](#_Toc8441_WPSOffice_Level1)

[CHAPTER TWO](#_Toc22496_WPSOffice_Level1) [12](#_Toc22496_WPSOffice_Level1)

[LITERATURE REVIEW](#_Toc1093_WPSOffice_Level1) [12](#_Toc1093_WPSOffice_Level1)

[2.1 Introduction](#_Toc244_WPSOffice_Level1) [12](#_Toc244_WPSOffice_Level1)

[2.2 Home automation](#_Toc26047_WPSOffice_Level1) [12](#_Toc26047_WPSOffice_Level1)

[2.4 Database](#_Toc17799_WPSOffice_Level1) [15](#_Toc17799_WPSOffice_Level1)

[2.5 Biometrics](#_Toc24485_WPSOffice_Level1) [15](#_Toc24485_WPSOffice_Level1)

[2.6 Microcontroller](#_Toc30998_WPSOffice_Level1) [16](#_Toc30998_WPSOffice_Level1)

[2.7 Review of Related Work](#_Toc30934_WPSOffice_Level1) [16](#_Toc30934_WPSOffice_Level1)

[CHAPTER THREE](#_Toc19217_WPSOffice_Level1) [18](#_Toc19217_WPSOffice_Level1)

[METHODOLOGY AND MATERIALS](#_Toc32477_WPSOffice_Level1) [18](#_Toc32477_WPSOffice_Level1)

[3.1 Introduction](#_Toc18220_WPSOffice_Level1) [18](#_Toc18220_WPSOffice_Level1)

[3.2 HARDWARE REQUIREMENT](#_Toc22936_WPSOffice_Level1) [18](#_Toc22936_WPSOffice_Level1)

[3.3 Firebase configuration on android](#_Toc23804_WPSOffice_Level1) [22](#_Toc23804_WPSOffice_Level1)

[3.3 Arduino and ESP8266 Setup](#_Toc27844_WPSOffice_Level1) [23](#_Toc27844_WPSOffice_Level1)

[CHAPTER FOUR](#_Toc20930_WPSOffice_Level1) [25](#_Toc20930_WPSOffice_Level1)

[RESULTS AND DISCUSSION](#_Toc32562_WPSOffice_Level1) [25](#_Toc32562_WPSOffice_Level1)

[4.1 Introduction](#_Toc21847_WPSOffice_Level1) [25](#_Toc21847_WPSOffice_Level1)

[4.2 RESULT](#_Toc7290_WPSOffice_Level1) [25](#_Toc7290_WPSOffice_Level1)

[Fig 4.3 a](#_Toc31626_WPSOffice_Level1) [27](#_Toc31626_WPSOffice_Level1)

[CHAPTER FIVE](#_Toc19808_WPSOffice_Level1) [28](#_Toc19808_WPSOffice_Level1)

[CONCLUSION AND RECOMMENDATION](#_Toc15062_WPSOffice_Level1) [28](#_Toc15062_WPSOffice_Level1)

[5.1 Conclusion](#_Toc19772_WPSOffice_Level1) [28](#_Toc19772_WPSOffice_Level1)

[5.2 Limitation](#_Toc5052_WPSOffice_Level1) [29](#_Toc5052_WPSOffice_Level1)

[5.4 Recommedation for further work](#_Toc23255_WPSOffice_Level1) [29](#_Toc23255_WPSOffice_Level1)

[References](#_Toc15010_WPSOffice_Level1) [30](#_Toc15010_WPSOffice_Level1)

[APENDIX](#_Toc11553_WPSOffice_Level1) [31](#_Toc11553_WPSOffice_Level1)

**TABLE OF FIGURES**

[Table 1 Relay Fig3.1 19](#_Toc1662)

[Table 2 NodeMCU Fig 3.2 20](#_Toc19827)

[Table 3 Jumper wire FIg 3.3 20](#_Toc15165)

[Table 4 Breadboard Fig 3.4 21](#_Toc1075)

[Table 5 Node MCU Fig3.5 25](#_Toc4388)

[Table 6 Basic Circuit diagram of Node MCU connection. Fig 4.1 27](#_Toc20638)

[Table 9 Home page Fig 4.2 c 28](#_Toc3243)

[Table 10 Final Setup2 Fig 4.3 b 28](#_Toc4241)

[Table 11 Final Setup2 Fig 4.3 b 29](#_Toc29291)

[Table 12 Firebase1 Fig 4.4 a 29](#_Toc14498)

[Table 13 Firebase2 Fig 4.4b 30](#_Toc22263)

**CHAPTER ONE**

**INTRODUCTION**

* 1. **Background of study**

Research on Home Automation devices have been going on for quite some time. These research have produced smart thermostats, switches and wireless controllable lights like Philips Hue. But these devices are mostly out of reach of the general population mainly because of its high cost.

Introduction these smart electrical lights, switches have enabled us to control electrical appliances through our phones. These are low cost devices that can be used to control various electronics that have built-in microcontrollers and wireless connectivity. But sadly, traditional electronics do not contain either microcontrollers, nor have wireless capabilities.

Our solution enables users control to everyday electronics just adding some additional simple components to their existing electronic switches and controls through Android phones.

1.2 Motivation

Due to the advancement of wireless technology, there are several different of connections are introduced such as GSM,WIFI, ZIGBEE, and Bluetooth. Each of the connection has their own unique specifications and applications. Among the four popular wireless connections that often implemented in HAS project, wireless Esp-8266 module is being chosen to increase the range with compare to bluetooth. Android, by Google Inc. provides the platform for the development of the mobile applications for the Android devices. There are so many android applications available in google play store or we can design android application according to our requirement. Here, we will be designing our android application using Android studio.

**1.3 Problem statement**

The problem is that, to use the existing solutions, users need to buy expensive electronics which often include single use lights and socket adapters with built in wireless connectivity which needs to be thrown out after it fuses or breaks down .

In case of switches, users need to replace switches with expensive ones to gain the ability to control them wirelessly. In case of sockets, users need to add additional sockets over wall sockets and it does not provide individual control over each device connected via the

1.4 Aims and Objectives

## The objective of this project is to provide a design or pattern that can be used to implement real-time remote control on everyday electronics. The goal of this project is to implement a secured real-time control of home appliances and other electrical components through Mobile Phones.

**1.5 Project organization**

The paper follows a very common organization pattern. The first part starts with the introduction of the project. The second part describes over related work in the same field. The system model has been described in the third part containing description of different components of the system. The fourth part describes the system in action or the system implementation including the implementation and features of Android application, arduino uno and Firebase web service. The technical and non-technical limitations are described in in the fifth part. The paper concludes with hints of future expansion ideas and pathways and lastly, a short conclusion. Additional system diagrams, illustrations and code samples are added as appendix.

**CHAPTER TWO**

**LITERATURE REVIEW**

2.1 Introduction

This chapter reviews the related literature of this study. It explains what Home automation is. It also defines what android Application, Flutter SDK, Firebase,Node MCU.It also covers biometric is and how it operates.

**2.2 Home automation**

Home automation is similar to digital home, smart home, e-home and intelligent household. They all mean a high living condition with several smart devices. It is the residential extension of automation which is using telecommunication technology, computer technology and automation technology to give the user a better living environment, security and comfort. It helps people to reduce domestic working and household management by its automation and monitoring system.

### 2.2.1 History of home automation

Early home automation began with labour-saving machines. Self-contained electric or gas powered [home appliances](https://en.wikipedia.org/wiki/Home_appliances" \o "Home appliances) became viable in the 1900s with the introduction of [electric power distribution](https://en.wikipedia.org/wiki/Electric_power_distribution" \o "Electric power distribution) and led to the introduction of [washing machines](https://en.wikipedia.org/wiki/Washing_machine" \o "Washing machine) (1904), [water heaters](https://en.wikipedia.org/wiki/Water_heater" \o "Water heater) (1889), [refrigerators](https://en.wikipedia.org/wiki/Refrigerator" \o "Refrigerator), [sewing machines](https://en.wikipedia.org/wiki/Sewing_machine" \o "Sewing machine), [dishwashers](https://en.wikipedia.org/wiki/Dishwasher" \o "Dishwasher), and [clothes dryers](https://en.wikipedia.org/wiki/Clothes_dryer" \o "Clothes dryer).

In 1975, the first general purpose home automation network technology, [X10](https://en.wikipedia.org/wiki/X10_(industry_standard)" \o "X10 (industry standard)), was developed. It is a communication protocol for electronic devices. It primarily uses [electric power transmission](https://en.wikipedia.org/wiki/Electric_power_transmission" \o "Electric power transmission) wiring for signalling and control, where the signals involve brief [radio frequency](https://en.wikipedia.org/wiki/Radio_frequency" \o "Radio frequency) bursts of [digital data](https://en.wikipedia.org/wiki/Digital_data" \o "Digital data), and remains the most widely available By 1978, X10 products included a 16 channel command console, a lamp module, and an appliance module. Soon after came the wall switch module and the first X10 timer.

By 2012, in the United States, according to ABI Research, 1.5 million home automation systems were installedAs per research firm Statista more than 45 million smart home devices will be installed in U.S. homes by the end of the year 2018

### 2.3.2 Android Framework

Android is one of an Open source platform. It is made by Google and owned by Open Handset Alliance. It is structured with objective “accelerate innovation in mobile” As such android has taken over a field of mobile innovation. It is definitely free and open platform that differs hardware from software that runs on it. It results for considerably more gadgets be running a similar application. Likewise, it gives plausibility of friendlier condition for developers and consumers. Android is a finished programming bundle for mobile devices. Since the beginning android group offers the developing kit (tool and frameworks) for creating mobile applications fast and simple as could be expected under the circumstances. In a few cases you do not specially need an android phone however you are free to have one. It can work right out of the box, obviously users can modify it for their specific needs. For manufacturers it is ready and free solution for their devices. Except specific drivers’ android community provides everything else to create their devices

**2.3.3 Dart Programming language**

Dar**t** is a general purpose programming anguage originally developed by Google in 2011 and later approved as a standard by Ecma (ECMA-408). It is used to build web,server, desktop, and mobile applications.Dart is an object oriented, class defined, garbage collected language using a C-style synatax that transcompiles optionally into Javascript. It supports interface, mixins, abstract classes, reified generics, [s](https://en.wikipedia.org/wiki/Static_typing" \o "Static typing)tatic typing, and a sound type system.

**2.3.4 SDK**

A software development kit (SDK or devkit) is typically a set of [software development](https://en.wikipedia.org/wiki/Software_development" \o "Software development) tools that allows the creation of [applications](https://en.wikipedia.org/wiki/Application_software" \o "Application software) for a certain [software](https://en.wikipedia.org/wiki/Software" \o "Software) package, [software framework](https://en.wikipedia.org/wiki/Software_framework" \o "Software framework), hardware platform, [computer system](https://en.wikipedia.org/wiki/Computer_system" \o "Computer system), [video game console](https://en.wikipedia.org/wiki/Video_game_console" \o "Video game console), [operating system](https://en.wikipedia.org/wiki/Operating_system" \o "Operating system), or similar development platform. To enrich applications with advanced functionalities, advertisements, push notifications,and more, most app developers implement specific software development kits.

**2.3.5 Flutter**

Flutter is an open source mobile reactive development framework created by Google officially launched in 2017. It is used to develop applications for Android and Ios, as well as being the primary method of creating applications for Google new Operating system called fuchsia.Flutter apps are written in the Dart language and make use of many of the language's more advanced features On Android, and on Windows, macOS and Linux via the semi-official Flutter Desktop Embedding as well Humming bird project project for web, Flutter runs in the Dart virtual machine which features a just in time execution engine. Due to App Store restrictions on dynamic code execution, Flutter apps use ahead of time compilation on IOS.A notable feature of the Dart platform is its support for "hot reload" where modifications to source files can be injected into a running application. Flutter extends this with support for stateful hot reload, where in most cases changes to source code can be reflected immediately in the running app without requiring a restart or any loss of state.This feature as implemented in Flutter has received widespread praise.

**2.4 Database**

A database is an organized collection of [data](https://en.wikipedia.org/wiki/Data_(computing)" \o "Data (computing)), generally stored and accessed electronically from a computer system. Where databases are more complex they are often developed using formal [design and modeling](https://en.wikipedia.org/wiki/Database" \l "Design_and_modeling) techniques.

2.5 Biometrics

Biometrics are a way to measure a person’s physical characteristics to verify their identity. These can include physiological qualities, for example, fingerprints and eyes, or behavioral characteristics such as the unique way you complete a security-authentication puzzle. To be helpful, biometric data must be unique, permanent and collectible. Once measured, the information is compared and matched in a database.

Fingerprint recognition looks for the unique patterns of ridges and valleys that are present in an individual’s fingerprint. These patterns are unique to every individual and thus help to identify individuals from an entire population. Fingerprints are inherent to individuals and can neither be lost nor stolen which makes it highly accurate and reliable.

### 2.5.1 Authentication

This is the process of recognizing a user’s identity. It is the component of linking an incoming request with a set of identifying credentials. The credentials provided are contrasted with those on a record of a database of the authorized user’s information on a local operating system or within an authentication server.

**2.6 Microcontroller**

A microcontroller (MCU for microcontroller unit) is a small [computer](https://en.wikipedia.org/wiki/Computer" \o "Computer) on a single [integrated circuit](https://en.wikipedia.org/wiki/Integrated_circuit" \o "Integrated circuit). In modern terminology, it is similar to, but less sophisticated than, a [system on a chip](https://en.wikipedia.org/wiki/System_on_a_chip" \o "System on a chip) (SoC); an SoC may include a microcontroller as one of its components. A microcontroller contains one or more [CPUs](https://en.wikipedia.org/wiki/Central_processing_unit" \o "Central processing unit) ([processor cores](https://en.wikipedia.org/wiki/Processor_core" \o "Processor core)) along with [memory](https://en.wikipedia.org/wiki/Computer_memory" \o "Computer memory) and programmable [input/output](https://en.wikipedia.org/wiki/Input/output" \o "Input/output) peripherals.

2.7 Review of Related Work

## Over the last decade, smartphone apps have allowed us to advance the progress of systems and processes that hadn’t changed in years. Twenty years ago, no one ever thought you’d be able to take your phone out of your pocket at work and view the real-time security footage of your home. Today however, that’s only one of the benefits of turning your home into a smart home using home automation. Now, millions of people around the world are also using home automation apps to do things like reduce their home energy usage and run their entertainment systems.

### **Samsung SmartThings:** This app goes beyond the capability of most home automation apps, not only allowing you to control devices such as cameras, locks and thermostats – but also giving you control over Samsung appliances like TVs, washing machines and air conditioners.

### **Sonos Controller:** If you are a music fanatic, today’s technology allows you to build the  epic smart entertainment system that you only dreamed about in the past. The Sonos Controller app connects wirelessly to Sonos speakers and allows you to control music throughout your home

**Philips Hue:** The Phillips Hue app allows you to control the entire range of [Philips Hue smart bulbs](https://www2.meethue.com/en-us/products.bulbs" \l "filters=STARTER_KITS_SU&sliders=&support=&price=&priceBoxes=&page=&layout=12.subcategory.p-grid-icon). Their Smart Bulbs can showcase over 16 million colors and can be dimmed or brightened remotely.

### **Nest:** Nest is known as one of the most innovative companies in the smart device sector. They create a variety of products that are meant to protect and automate your home. Currently, the Nest app focuses on developing devices such as smart cameras, smart doorbells, thermostats, alarm systems, and smoke + CO alarms.

**CHAPTER THREE**

METHODOLOGY AND MATERIALS

3.1 Introduction

This chapter describes the architecture of the system which consists of system components and how the systems developed. It discuss about the data flow diagram of the system, the flowchart of the system, the use case of the system and the tools of the system.

**3.2 HARDWARE REQUIREMENT**

[ESP8266 WiFi module-1](http://rees52.com/arduino-modules/381-rtc-ds3231-at24c32-iic-module-precision-real-time-clock-module.html?search_query=ds3231&results=8" \t "https://rees52.com/diy-arduino/_blank)

[1 way 5V Relay module-1](http://rees52.com/arduino-modules/1011-12v-light-operated-switch-photosensitive-resistance-and-relay-module-nb016.html" \t "https://rees52.com/diy-arduino/_blank)

[Mini breadboard(170 pin)-1](http://rees52.com/circuit-board/155-breadboard-mini-self-adhesive.html" \t "https://rees52.com/diy-arduino/_blank)

[Jumper wires MALE TO MALE   – 10 piece](http://rees52.com/arduino/1079-male-to-male-connector-jumper-wire-cable-for-breadboard-65-piece-25-cm-aa053.html" \t "https://rees52.com/diy-arduino/_blank)

[Jumper wires MALE TO FEMALE   – 10 piece](http://rees52.com/conectors/153-jumper-wire-male-to-female-rk010.html" \t "https://rees52.com/diy-arduino/_blank)

Two pin bulb holder-1(not included in kit)

Led bulb 220v – 1(not included in kit)

[Single stand wire 2m - 1](http://rees52.com/conectors/809-20-meter-multistand-wires-for-diy-electronics-projects-rc045.html" \t "https://rees52.com/diy-arduino/_blank)

**3.2.1 SOFTWARE REQUIREMENT**

Android Studio IDE

Flutter SDK

Firebase SDK

Arduino IDE

C/C++ programming language will be used to write the programme on the arduino

**3.2.2 Relay**

A relay is usually an electromechanical device that is actuated by an electrical current. The current flowing in one circuit causes the opening or closing of another circuit. Relays are like remote control switches and are used in many applications because of their relative simplicity, long life, and proven high reliability. Relays are used in a wide variety of applications throughout industry, such as in telephone exchanges, digital computers and automation systems. Highly sophisticated relays are utilized to protect electric power systems against trouble and power blackouts as well as to regulate and control the generation and distribution of power. In the home, relays are used in refrigerators, washing machines and dishwashers, and heating and air-conditioning controls.



Table 1Relay Fig3.1

**3.2.3 NodeMCU**

NodeMCU is an open source [IoT](https://en.wikipedia.org/wiki/Internet_of_Things" \o "Internet of Things) platform. It includes [firmware](https://en.wikipedia.org/wiki/Firmware" \o "Firmware) which runs on the [ESP8266](https://en.wikipedia.org/wiki/ESP8266" \o "ESP8266) [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi" \o "Wi-Fi) [SoC](https://en.wikipedia.org/wiki/System_on_a_chip" \o "System on a chip) from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the [Lua](https://en.wikipedia.org/wiki/Lua_(programming_language)" \o "Lua (programming language)) scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson and [SPIFFS](https://en.wikipedia.org/w/index.php?title=SPIFFS&action=edit&redlink=1" \o "SPIFFS (page does not exist)).

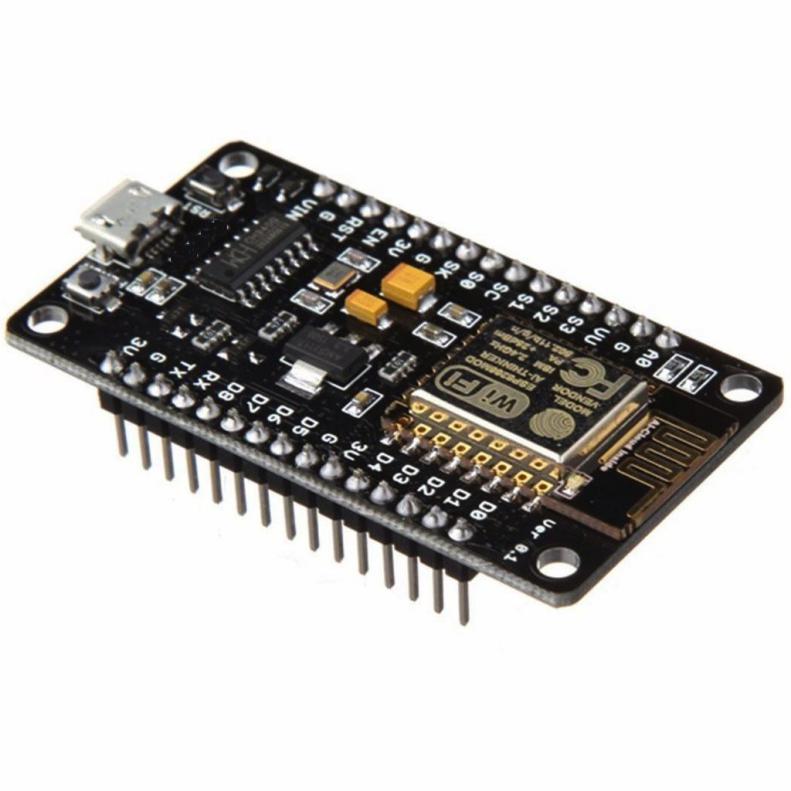


Table 2 NodeMCU Fig 3.2

**3.2.4 Jumper wire**

Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with [breadboards](https://blog.sparkfuneducation.com/what-is-a-breadboard) and other prototyping tools in order to make it easy to change a circuit as needed. Fairly simple. In fact, it doesn’t get much more basic than jumper wires.



Table 3 Jumper wire FIg 3.3

**3.2.5 Breadboard**

A breadboard is a solderless device for temporary prototype with electronics and test circuit designs. Most electronic components in electronic circuits can be interconnected by inserting their leads or terminals into the holes and then making connections through wires where appropriate. The breadboard has strips of metal underneath the board and connect the holes on the top of the board. The metal strips are laid out as shown below. Note that the top and bottom rows of holes are connected horizontally and split in the middle while the remaining holes are connected vertically.

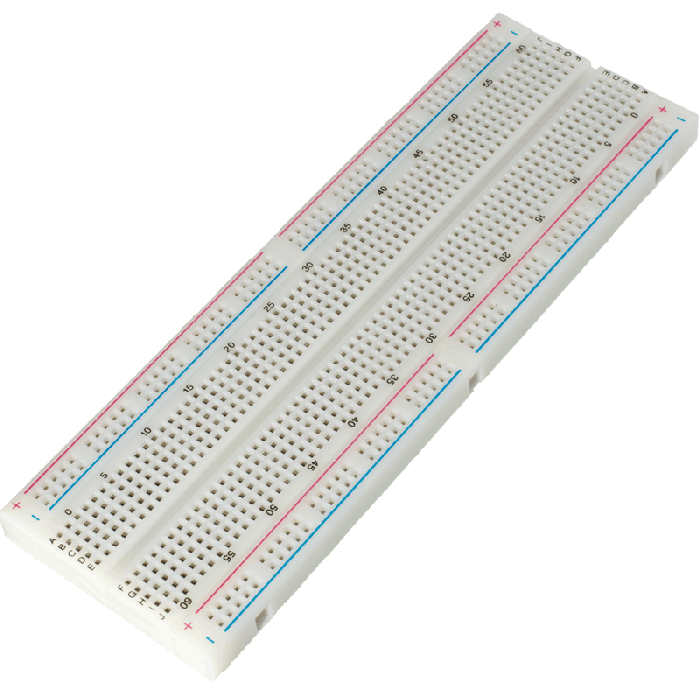


Table 4 Breadboard Fig 3.4

**3.2.6 ESP 8266**

The ESP8266 module is a self-contained System On Chip (SOC), which features an integrated TCP/IP protocol stack that allows you to add Wi-Fi capability to your projects. The module is usually mounted on circuit boards that break out the pins of the ESP8266 chip, making it easy for you to program the chip and to interface with input and output devices

# **3.2.7 CH340 Windows Driver**

#### CH340 IC is a low cost USB to TTL converter IC. [CH340g IC](http://www.dnatechindia.com/ch340-usb-ttl-ic.html) is used in [low cost SMD Arduino UNO](http://www.dnatechindia.com/arduino-uno-r3-smd-india-buy-low-cost.html" \t "https://www.dnatechindia.com/_blank) & [Arduino Nano boards](http://www.dnatechindia.com/Arduino-Nano-CH340.html" \t "https://www.dnatechindia.com/_blank). Many USB to Serial converter IC’s are available in the market but this is a low cost IC that is gaining popularity. USB to TTL converter modules are also available based on this IC. Even Node MCU IOT modules also have this IC.

**3.3 Firebase configuration on android**

### ****Step 1****: Create a Firebase project:

Before we can add Firebase to your Android app, we need to create a Firebase project to connect to our Android app by visiting [www.firebase.google.com](http://www.firebase.google.com)

### ****Step 2****: Register your app with Firebase

After you have a Firebase project, we can add your Android app to it. In the center of the [Firebase console's project overview page](https://console.firebase.google.com/), click the **Android** icon to launch the setup workflow.

### ****Step 3****: Add a Firebase configuration file

Add the Firebase Android configuration file to your app:

Click **Download google-services.json** to obtain your Firebase Android config file (google-services.json).

Move the config file into the module (app-level) directory of your app.

### ****Step 4****: Add Firebase SDKs to your app

1. To your module (app-level) Gradle file (usually app/build.gradle), add the dependencies for the Firebase products that you want to use in your app.
2. Sync your app to ensure that all dependencies have the necessary versions.

**3.3 Arduino and ESP8266 Setup**

In order to **setup your Arduino IDE** to work with your esp8266 arduino compatible module you need to make the following steps:

1. Connect your ESP8266-01 Module to PC
2. Open your [Arduino](https://www.amazon.com/gp/product/B00T4QZPNI/ref=as_li_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=B00T4QZPNI&linkCode=as2&tag=geek07f-20&linkId=50555e08b522c1aa1ab8b115789a807b" \t "https://www.geekstips.com/esp8266-arduino-tutorial-iot-code-example/_blank)IMG_256 IDE
3. Go to File -> Preferences
4. Add [this link](http://arduino.esp8266.com/stable/package_esp8266com_index.json) to Additional Board Manager
5. Go to Tools -> Board Manager
6. Find ESP8266 board set and activate it
7. Select Generic ESP8266 board from Tools->Boards
8. Choose your programmer COM port
9. We are ready to go!

**3.5 Arduino IDE ESP8266 board installation**

Now, to be able to download the program to your ESP-01 module, you first need to put your device in the **proper BOOT mode** (Download code from UART). ESP8266-01 have the following **boot modes**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MTDO / GPIO15** | **GPIO0** | **GPIO2** | **Mode** | **Description** |
| L | L | H | UART | Download code from UART |
| L | H | H | Flash | Boot from SPI Flash |
| H | X | X | SDIO | Boot from SD-card |

After resetting the module in Download code from UART you should see a message containing “boot mode:[1,6]” in the serial monitor, if you are on the correct baud rate. A wrong baud rate setting will display garbage text / characters or nothing at all. After that you should be able to upload your sketch to ESP8266. When upload is done, module should reset itself. Don’t forget to **pull HIGH the GPI0** or the module will get in Download mode **again** and you will not be able to see it working. The module can be rebooted at anytime by pulling REST pin to LOW. After each reset it will follow the boot sequence and program loading.

Once the **ESP8266** board is installed and activated in [Arduino](https://www.amazon.com/gp/product/B00T4QZPNI/ref=as_li_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=B00T4QZPNI&linkCode=as2&tag=geek07f-20&linkId=50555e08b522c1aa1ab8b115789a807b" \t "https://www.geekstips.com/esp8266-arduino-tutorial-iot-code-example/_blank)IMG_256 IDE, you will be able to include **all ESP WiFi libraries and examples** that comes with the package. The most used library is **ESP8266WiFi** which offers many implementation examples like **WiFiClient, WiFiServer, WiFiAccessPoint** etc. You can find allot of projects examples over the internet, I for example, found great ideas on [arduino.cc projecthub.](https://create.arduino.cc/projecthub/) Here is a simple **Arduino blink example** which you can use to test the esp module with the built in LED:

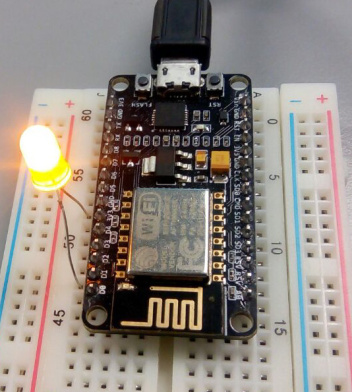


Table 5 Node MCU Fig3.5

**CHAPTER FOUR**

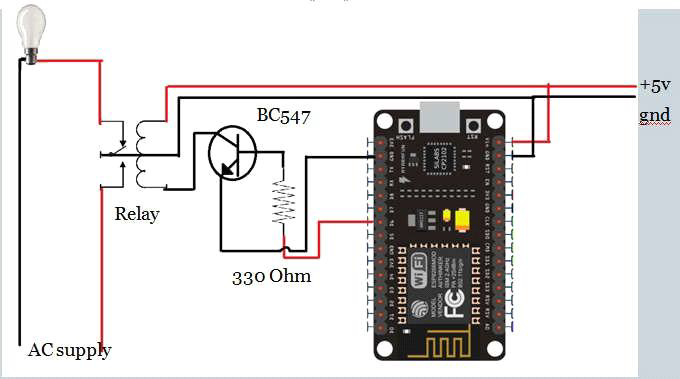
**RESULTS AND DISCUSSION**

4.1 INTRODUCTION

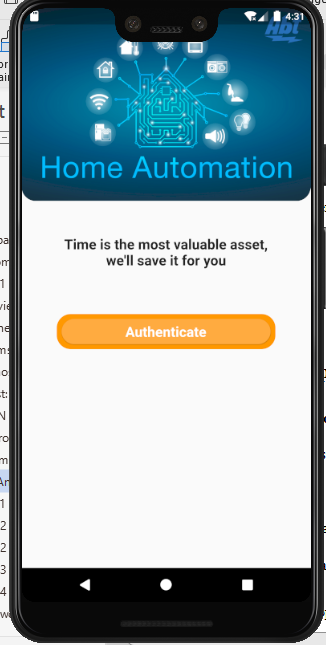
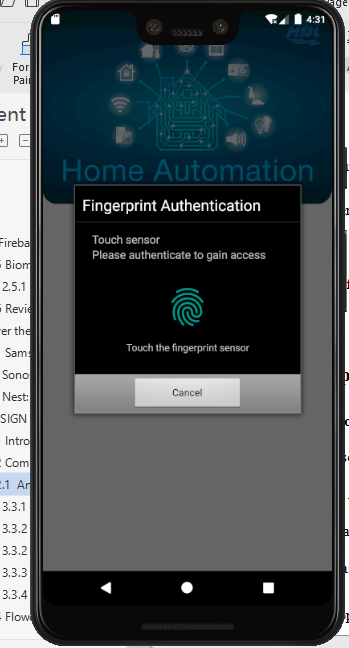
## This chapter put together all the components discussed in previous chapters to form a single circuit of the entire project, it also entails the results of the project

**4.2 RESULT**

Experimental results of system which is proposed in this paper are as below, Figure 5.1 depicts the circuit diagram of the overall system 5.2 (a)(b)(c) shows the mobile application part of the project, when the application is launched it shows an authentication page which allows the user to input their fingerprint ,if the regisered fingerprint matches with the inputed one it allows the user to homepage,which gives the user priviledge to toggle the switch button in the home screen as shown in figure below fig 5.2 shows hardware implementation of the android app to support home automation and actual appliances respectively. In designed android app, switch toogle is used to turn ON or OFF the Light 230v 50Hz electric bulb. Fig 5.4(a) shows the image of the firebase database at inital stage when the the switch is off, fig 5.4(b) shows the image of firebase database after the ON button on the mobile app is been pressed.



Basic Circuit diagram of Node MCU connection. Fig 4.1

Sign in page Fig 4.2a Fingerprint page Fig4.2b

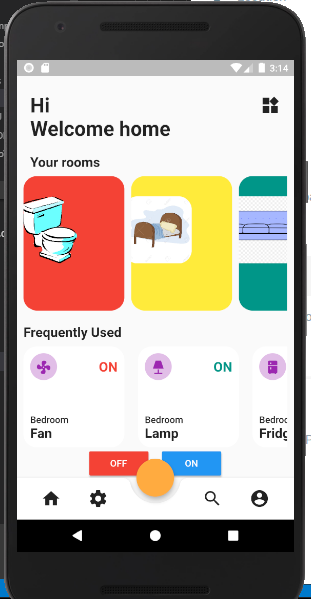
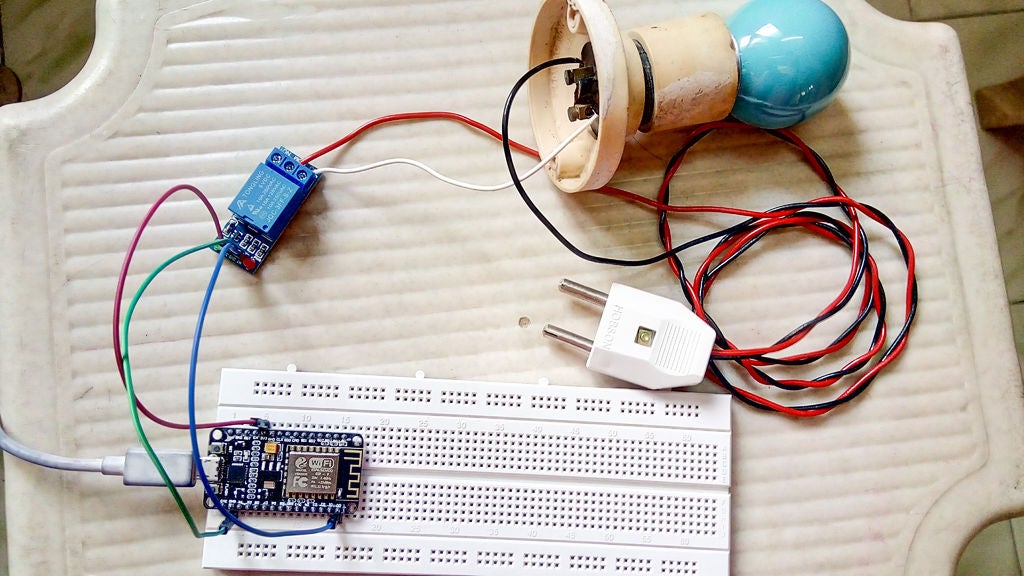


Table 9 Home page Fig 4.2 c



Final Setup2 Fig 4.3 b

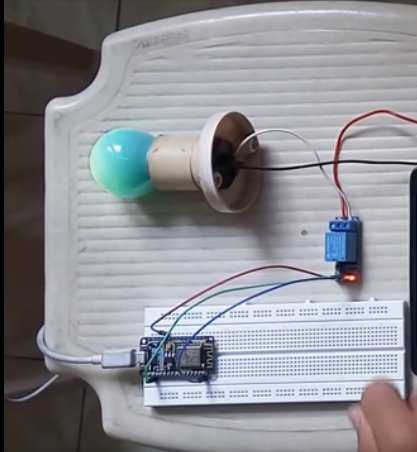
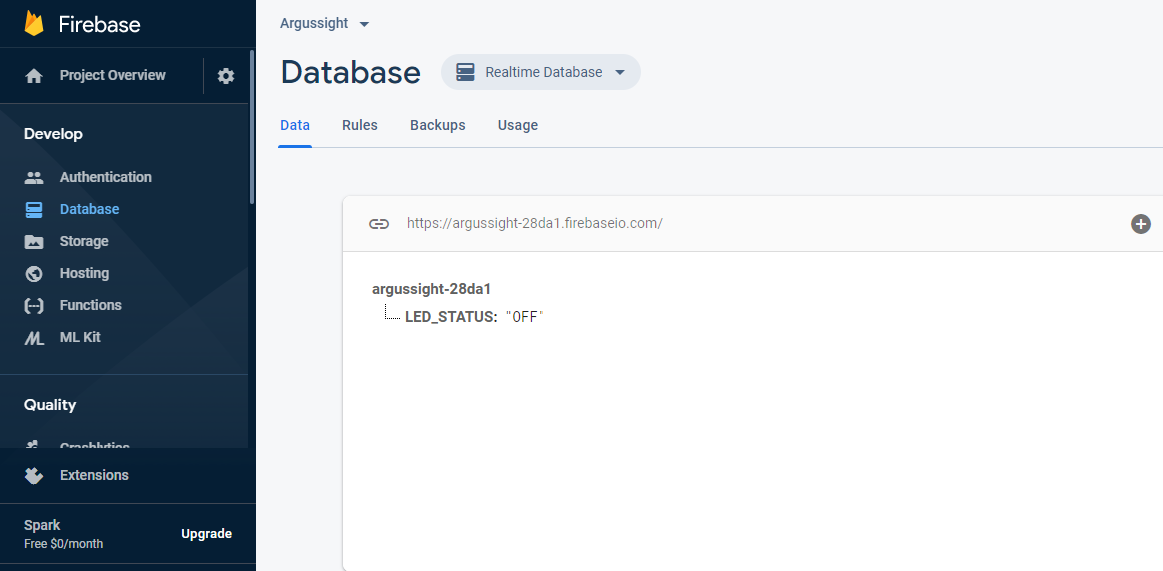
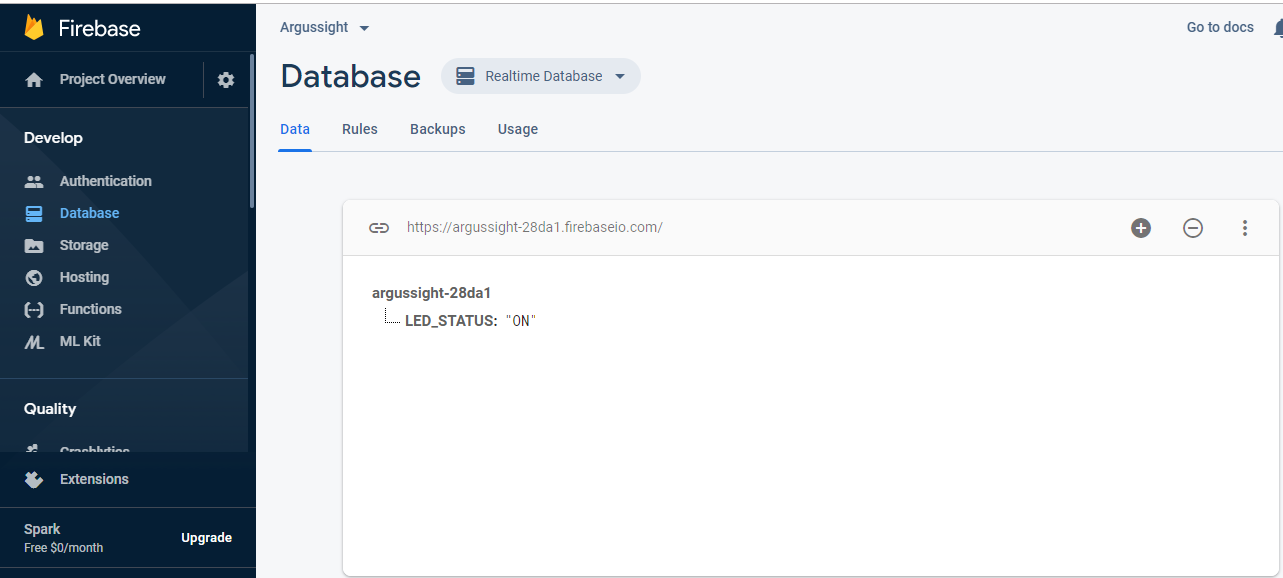
****

Table 11 Final Setup2 Fig 4.3 b

****

Firebase1 Fig 4.4 a

****

Firebase2 Fig 4.4b

**CHAPTER FIVE**

**CONCLUSION AND RECOMMENDATION**

**5.1 Conclusion**

Due to recent technological advancements, home automation systems will be soon in widespread use. One of the main issues that resist us from automating our home is the expensive cost of these smart equipment. Our designed system is a low cost alternative of the traditional home automation systems available in the market. We have designed a system that is developed using open source software and uses low cost hardware.

**5.2 Limitation**

The current technological limitations limit the ability to control all and any types of appliances. Since for our home and office appliances do not have built in micro controllers, controlling these electronic equipment cannot be possible. We can only turn on or off most of the appliances. Home Automation Systems often have door locks, thermostats and other devices that, is if abused my cause life threatening situations or cause huge damage to property. Since these home automation systems are mostly wirelessly connected, security is the biggest risk. Research on Security for IOT systems such as Home Automation System is undergoing. Still much research is needed for complete security of these systems. Non-technical limitations include slow comprehension of Smart and Automated Equipment. Besides these, people are usually reluctant to steep changes. Acceptance of Smart Home systems my take some more years.

**5.4 Recommedation for further work**

The current project can be extended in various ways. One possible direction would be to integrate voice interface to the system so that the system can be operated using voice only. One such solution would be to integrate Alexa Voice Service(Amazon.com, n.d.) by Amazon.com, Inc. After integrating with Alexa Voice Service, the system can be operated using Amazon Echo Devices and other devices that have implemented the service. Control of electronic components operated by voice would be a marvelous solution for physically disabled people. Another possible extension could be the implementation of automation in factories and manufacturing plants.

References

Amazon.com,”I. (n.d.). Alexa Voice Service (AVS). Retrieved from L. (2015a). Raspberry PI And Wifi Based Home Automation. In International Journal of Engineering Research and Applications (Vol. 11, pp. 57–60). https://doi.org/10.9790/2834-1103032530 Bhagyalakshmi, P., & Aravinda, N. L. https://doi.org/10.9790/2834-1103032530 Cooke, B. (n.d.). onoff. Retrieved from https://github.com/fivdi/onoff Google. (2011a). Firebase Realtime Database. Google. (2011b). Firebase Realtime Database. Retrieved from https://firebase.google.com/products/realtime-database.

application framework. Retrieved March 9, 2018, from https://expressjs.com/ Pandya, B., Mehta, M., & Jain, N. (2016). Android Based Home Automation System Using Bluetooth & Voice Command, 3–5. Patel, M. M., Jajal, M. A., & Dixita, B. (2015). http://ijiere.com/(X(1)S(rfsubrmlhygnnj3sckud21e4))/FinalPaper/FinalPaper201532874046379.pdf Piyare, R. (2013). Internet of Things : Ubiquitous Home Control”and Monitoring System using Android based Smart Phone. International Journal of Internet of Things, 2(1), 5–11. https://doi.org/10.5923/j.ijit.20130201.02 Raspberry Pi Retrieved March 9, 2018, from https://www.raspberrypi.org/ Saxena, S., Gupta, R., Varshney, S., Singh, S. P., & Kumari, S. (2016). Internet of Things based Home

**APENDIX**

#include <ESP8266WiFi.h> // esp8266 library

#include <FirebaseArduino.h> // firebase library

#define FIREBASE\_HOST "argussight-28da1.firebaseio.com" // the project name address from firebase id

#define FIREBASE\_AUTH "vMo88fg0aH7K41NArGsnpq9qsJ7BP8E2weqYNA4B" // the secret key generated from firebase

#define WIFI\_SSID "ibrahim" // input your home or public wifi name

#define WIFI\_PASSWORD "baksman73" //password of wifi ssid

String fireStatus = ""; // led status received from firebase

int led = 2; // for external led

void setup() {

Serial.begin(9600);

delay(1000);

pinMode(LED\_BUILTIN, OUTPUT);

pinMode(led, OUTPUT);

WiFi.begin(WIFI\_SSID, WIFI\_PASSWORD); //try to connect with wifi

Serial.print("Connecting to ");

Serial.print(WIFI\_SSID);

while (WiFi.status() != WL\_CONNECTED) {

Serial.print("\*");

delay(500);

}

Serial.println();

Serial.print("Connected to ");

Serial.println(WIFI\_SSID);

Serial.print("IP Address is : ");

Serial.println(WiFi.localIP()); //print local IP address

Firebase.begin(FIREBASE\_HOST, FIREBASE\_AUTH); // connect to firebase

Firebase.setString("LED\_STATUS", "OFF"); //send initial string of led status

//Firebase.setBool("LED\_STATUS", true);

}

void loop() {

fireStatus = Firebase.getString("LED\_STATUS"); // get ld status input from firebase

if (fireStatus == "ON") { // compare the input of led status received from firebase

Serial.println("Led Turned ON");

digitalWrite(LED\_BUILTIN, LOW); // make bultin led ON

digitalWrite(led, HIGH); // make external led ON

}

else if (fireStatus == "OFF") { // compare the input of led status received from firebase

Serial.println("Led Turned OFF");

digitalWrite(LED\_BUILTIN, HIGH); // make bultin led OFF

digitalWrite(led, LOW); // make external led OFF

}

else {

Serial.println("Wrong Credential! Please send ON/OFF");

}

}